RU Staying

Software Engineering
01:332:452
Report 1

By Group #11
Keya Patel
Zain Sayed
Mohammed Sapin
Purna Haque
Nga Man (Mandy) Cheng
Rameen Masood
Shilp Shah
Mathew Varghese
Thomas Tran
Eric Zhang

Github: https://github.com/mohammedsapin/RUStaying
Submission Date: Feb 24, 2019

Table of Contents

Chapter 1

Customer Statement of Requirements

1.1 Problem Statement

Customer:

Tommy had a flight booked for American Airlines on Thursday. He wanted to enjoy a peaceful trip to Princeton. He booked a package deal on Expedia for flight and hotel. He had planned to stay at a local Hilton hotel located in Princeton, NJ. Tommy was having a terrible time at the start of his vacation. His flight was delayed, TSA was giving him a hard time, and a baby was crying the entire flight. By the time he landed he had to wait 40 minutes for a runway to become available, another 40 minutes for his luggage to get to the carousel, and another 40 minutes for an Uber pool where he shared it with a slightly “eccentric” family.

By the time he got to the hotel, Tommy did not want to deal with another person, nor did he want to wait in line to be checked in. But to his disappointment, when he got to the hotel, the line was extremely long, held up by someone complaining at the front desk. After he finally got to the front of the line, checked in, and asked for a bell boy, he found himself again impatiently waiting for someone to come help him carry his heavy luggage up to his room. However, a bell boy was nowhere to be found, and Tommy had no choice but to wait in the lobby just for one to show up.

When he finally got to his room, all he wanted to do was rest from his terribly long day, but he could not even get into his room due to his key card malfunctioning. So he had to go back to the front desk and wait in line again for a new key. Once he finally got a new key and was able to enter his room he realized his sheets had not been changed, and the toilet wouldn't flush. He tried calling room service on the phone but the lines were busy, so again he had to go to the front desk and report the issues. However, the front desk told him that maintenance was busy, and that Tommy would have to wait for staff to check if another room was available.

As Tommy was hungry, he figured that he would let the hotel staff figure out his room situation while he grabbed a bite to eat. He walked over to the hotel restaurant, sat down at a table and relaxed in his chair. He figured that after a meal he would be in better spirits. However, when the waiter handed him his menu, to his horror he discovered that the hotel had no vegan options. Tommy, a devout believer in veganism, found his efforts to relax again thwarted by a lack of information readily available for him, and attempted to leave the hotel to find outside dining options. But as soon as he stepped outside, he was immediately hit with a wall of torrential rainfall. Now fuming, he decided to check on the room status, and to his pleasant surprise, the hotel had been able to quickly move his belongings to a new room. Entering, he checked the toilet, sink, and bed for any signs that his new room was dysfunctional. Satisfied, he laid in his bed, ready to finally get a good night's rest after such as stressful day. However, he quickly discovered that sleeping would not be an option, as the hot and humid summer night caused a profusion of sweat to begin forming all over his body. He quickly reached for the air conditioning remote, and turned it on to its lowest setting. To his dismay, the AC began sputtering hot, foul smelling air into his room. At this point, Tommy was ready to cut his

vacation short and go home. After again going to the front desk and finding a new room, Tommy finally fell into a nightmare riddled sleep full of bathroom malfunctions and dirty bed sheets. When he awoke, he felt surprisingly refreshed with a clear purpose. He gathered all of his belongings and promptly checked out of the hotel. He had decided to go back home after all. On his way out, he asked if the hotel had any customer feedback forms he could fill out. After being told no, he happily yelled at the staff for an hour before departing for his flight home.

From Tommy's experience we can see that many issues can arise in a hotel. This could be do to poor hotel management, influx of many guests during vacation weekends, and many other things. The hotel was able to provide Tommy with solutions for each of his problems, however it was very inconvenient and took a long time. Tommy would have rather had his solutions addressed or solved at a timely manner. With our application we are trying to solve these issues and help people like Tommy have a better guest experience at hotels. Through the app Tommy can check in beforehand and receive a digital copy of his key on his phone. Before he enters the hotel he can even request for his room to have clean sheets, working appliances, etc. All of this saves Tommy time from waiting on the phone or going down to the front desk each time to solve an issue. Through the app we are enhancing the customers ability to have self service and do things on their own. Hotels will still operate the same way and guests can choose not to use the app and talk directly to the hotel staff. We are targeting a group of people that are more tech savvy and are part of the revolutionizing era of self-service technology. A typical guest through the app can request a car, report how many luggages he has, request amenities for his room, preview menus, order food, and even access the concierge. At the end of a users stay the app asks for feedback and ways to improve service.

Hotel Staff:

Kenny, the hotel manager, is tired from a long, endless night of checking people into the Varghese Hotel. As he stands at the concierge desk, typing furiously away trying to check-in people waiting, he is getting more stressed. Kenny tries his best to meet every guest with a smile and a friendly greeting, however, it becomes difficult when guests come in exhausted and irritated from their travels. Currently, it's 5 PM on a Friday night and there are 15 guests waiting impatiently to get checked-in. Kenny, thinking to himself, wishes that there was a way all of these problems could go away. If only there was an efficient way to check people in without angry customers yelling at him. As Kenny works tirelessly to make sure all customers are shown to their rooms, he realizes there are not enough bellboys for guests' luggage. The guests, realizing there is no bellboy, furiously just takes their luggage to their rooms themselves. Their stay at the hotel is not starting on a good note. As Kenny takes a quick moment to compose himself, his break is hurried with ring of the front desk phone. The complaints begin to pour in. Guests are filing maintenance requests, bedding changes, issues with the room keys, and much more. Although I have my hotel managing staff to help me, all the complaints are starting to pile up and guests are growing impatient. We are confident we will resolve all the issues, but not in a timely manner. On top of this, the technology we use at the hotel has many glitches. We try our best to keep track of room usage and just general monitoring, however our tech is outdated and slow. We need to make advancements and improve the way we manage the hotel.

App Solution (Manager/Staff):

- Manager can view assignments of digital keys for security purposes

- Bellboys will get a push notification when service is needed
- Manager can see which rooms are occupied and pull up the information of guest staying in that room, and the requests the guest may have listed
- Driving services/taxi will get notifications for scheduling appointments
- Maintenance staff will be notified for requests
- Restaurant staff will be able to see orders which have been requested

With the app (Solution):

- Put in luggage information through the app to make sure bellboy service is on time
- Digital Key to avoid problems with the physical key plus skip the LONG lines
- Put in urgent maintenance request (broken toilet, phone, AC), no need to call
- Ask for room cleaning (Dirty bed sheets) , no need to call
- Request for a another room, no need to call
- Check for restaurant hours
- Check the restaurant menu through the app and include dietary restrictions
- Eliminate wait times through check in with the app, and requests made in the app
- Feedback form
- Request a complimentary car service for pickups and drop-offs

Although Tommy was having a horrible course of events, these could have all been avoided through the use of the hotel app, RU Staying. Through the app, Tommy had the option to request a bellboy upon his arrival and even say how many luggages he will have when he does arrive. In addition, the digital key that is available through the app would have allowed Tommy

to avoid any malfunctions with a physical card key. These are just a few of many features that the app will be able to provide to guest who are seeking a more enjoyable and convenient stay.

1.2 Glossary of Terms

Bellboy - A staff member who helps guests in moving their luggage to their rooms, and provides any additional assistance that may be asked of them.

Concierge - A staff member who is responsible for answer any questions guests may have pertaining to anything from hotel policy to nearby points of interest.

Maid - A staff member whose primary responsibility is to clean the guests' rooms.

Maintenance request - Process by which a guest can request assistance when dealing with a technical or mechanical issue in their room

Reservation - Process by which a guest will reserve an available room and pick out other amenities or logistics

Check-In - Process by which a guest will confirm their booking information and payment methods and be allowed to move into their hotel room to begin their stay.

Digital Room Key - A digital card that, after booking a room, the guest will receive on their phones after checking in. It will allow them access to their rooms and other guest-only areas in the hotel such as the pool or gym.

Physical Room Key - A physical card version of the digital room key- it will allow guests access to their rooms and other guest-only areas in the hotel such as the pool or gym.

Creating an account - A process by which a guest creates a username and password to become associated with the hotel, and allow their information to be quickly accessed by hotel staff.

Guest - A person who has booked a room, checked in, and is now living in the hotel within his reserved duration

Valet Parking - A process by which a bellboy or another staff member will be given a guest's keys and drive the guest's vehicle to the hotels parking garage. The guest can also request for their car to be driven back to the front entrance if they wish to leave.

Feedback/customer satisfaction - Through a form we will gauge how much customers either enjoyed or did not enjoyed living at our hotel. This can be measured by any complaints or complements they may have, as well as any additional comments that we will read.

Luggage - Any amount of baggage that guests bring with them to keep during their stay.

Bedding - Additional room items such as bed sheets, pillows, blankets, towels, etc.

Early departure - Hotel policy that allows guests to leave before their original departure date.

Connecting room - 2 rooms in our hotel connected by a door. These types of rooms can be specially reserved by the guest.

Chapter 2

Function System Requirements

2.1 User Stories

REQ	Priority Weight	Description
REQ - 1	2	System will allow users to create an account and register with email and password
REQ - 2	2	System will allow users to login to their account with a unique username and password
REQ - 3	6	System will allow users to check-in for their room
REQ - 4	6	System will allow users to make reservations
REQ - 5	4	System will allow users to call room service.
REQ - 6	8	System will keep track of available / unavailable rooms
REQ - 7	5	System will allow users to make maintenance requests
REQ - 8	3	System will verify user account by checking with the database
REQ - 9	2	System will allow users to input how many bags they have in order to call the bellboy
REQ - 10	7	System will provide the user with a digital room key.
REQ - 11	1	System will provide the user with additional hotel information

		and FAQ.
REQ -12	3	System will allow guests to use the car service.
REQ - 13	10	System will be able to keep track of room usage and general hotel activity
REQ - 14	1	System will ask for feedback from the guest upon checkout
REQ - 15	3	System will allow users to easily communicate with staff
REQ - 16	6	System will allow users to check-out of their room
REQ - 17	1	System will provide an automated concierge service to answer questions

2.2 Non-Functional System Requirements

REQ	Priority Weight	Description
REQ - 18	6	The app will have a simple, responsive user interface (see details below about more specific requirements)
REQ - 19	1	The system should allow more than one user to access and use the application
REQ - 20	1	The system will use SQLite for data maintenance
REQ - 21	2	The system should provide error notifications to let the user know there is an issue

2.2.1 Functionality

The goal of this app is to become widely used by all the guests that stay at the hotel. Overtime, we expect a growing user base because more people will visit the hotel and revisit because of the convenience we provide through the app. To handle the growing user base, we will be storing our information using SQLite database. Since most of the data we are storing is user information and the use of the hotel services, we only needed a simple and functional tool. Data sorting and analysis tools are not needed so we decided against Postgresql. Since the app is on Android, it is already portable and usable by most devices in the world. For security, we will be protecting passwords by hashing when storing them in the database.

2.2.2 Usability

As mentioned, the app is being built on the Android platform so it is very accessible by many users. To have a modern app design that is aesthetic and responsive, we will be following Google's Material Design tools. This will save us a lot of time when creating the user interface because all the tools are built and we only have to implement them.

2.2.3 Reliability

We will be thoroughly debugging and testing our application to ensure that it does not crash at any point during use. Throughout our multiple builds, we will be continually updating our application to fix any issues that arise in runtime. This will help cut down on the possibility of failure if and when an actual customer is using it. Additionally, we will be backing up all the data saved onto a separate database table to ensure data is not lost or corrupt. We will also be making sure that customers will not have access to the managerial aspect of the app and vice versa to prevent any security risks or issues.

2.2.4 Performance

For the purposes of this app, we will not be managing a very large set of data, so the performance that SQLite provides is acceptable. The app is more focused on functionality and Android Studio is a reliable application that allows us to scale the app easily.

2.2.5 Sustainability

When implementing the requirements of the app, we will heavily use the features of OOP to maintain modularity and scalability. Also, we will use the Android Studio debugging tools to ensure an error-free experience for the user. Of course, errors can still occur so we will provide notifications in the app of the errors that occur. This way, while testing the app, we can see the type of errors and where they occur to try to fix them.

2.3 On-Screen Appearance Requirements

The on-screen requirements are specific to the user interface and were designed to provide a simple and convenient experience. We will be splitting up the user interface in two main parts, the first part is a page that provides the necessary information and the second part is a navigation bar that is consistent across all pages. This is a common modern design for many applications as it is simple and allows the user to navigate the app easily.

REQ -	Priority Weight	Description
REQ - 20	1	The app will have a navigation bar across the bottom to switch easily between the main features (ex: Home, Check-in, Guest Services, User Profile etc…).
REQ - 21	1	The Home button on the navigation bar will lead to a page where the user can explore more about the hotel, nearby points of interests, and FAQ.
REQ - 22	1	The User Profile tab will provide the users personal information and details about their hotel stay.
REQ - 23	1	Another tab will allow the user to Check-In or Make a

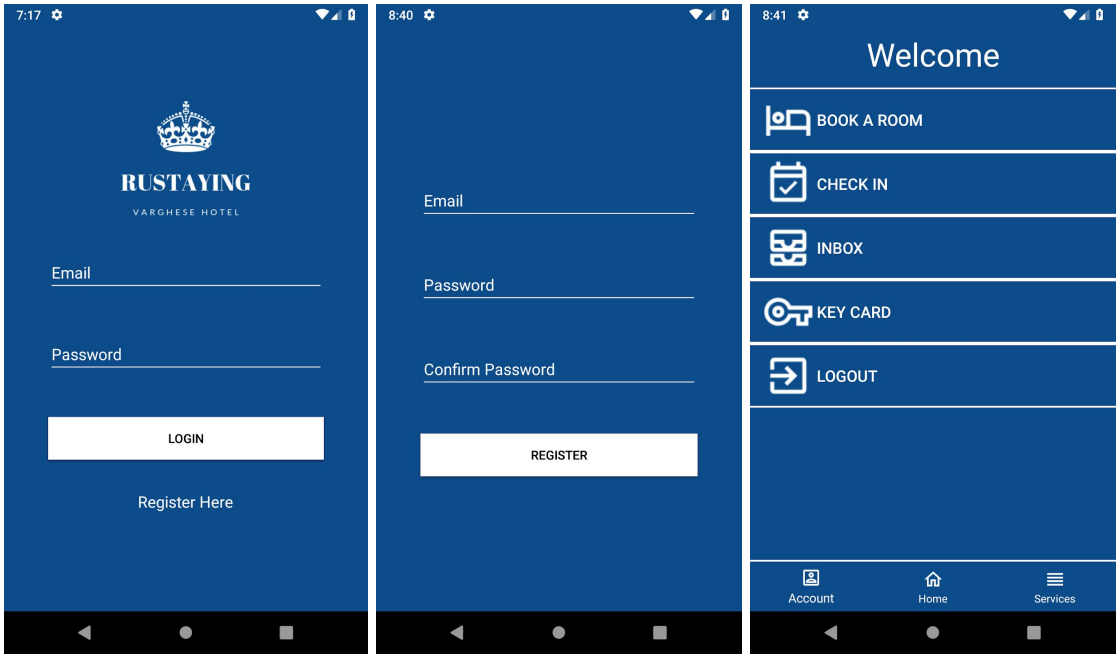
		Reservation. Upon clicking on these, it will lead to another page where the user can actually perform the action.
REQ - 24	1	We have a lot of services that the user can request so we will have one tab that lists all the services and from there the user can select one (see UI diagrams below).
REQ - 25	3	Within the Request Hotel Service tab, some of the services listed are: Concierge, Room Service, Valet & Travel Service, Call Bell Boy, Room Maintenance, and the Feedback Form. This is the main tab where the user can access all the services.

2.3.1 App User Interface Templates

Guest Services

Maintenance Request Page (from Guest Services)

Login Page



These three images are the final design that we are planning to implement for our application based on the mock-ups we created. The first image displays the page a user sees when he first opens our app. If a user does not have an account they can click Register Here and will be navigated to the another page as displayed by the second image. When a user successfully logs in to the app the first page they see is the welcome page as displayed by the third image. From here a user will be able to access all the services our app provides for them.

Chapter 3

Functional Requirements Specification

3.1 Stakeholders

The main stakeholders of this app are hotel owners and managers. The app is commissioned by the owners of the hotel in an attempt to expand the hotel brand and earn more

profit. With our design, this app can significantly improve the efficiency of the hotel services, which in turn increases the number of guests, therefore increasing profit. As investors, the hotel owners are certainly interested in the growth and success of the app. The hotel managers are also stakeholders because they control the day-to-day activities of the hotel and it is up to them to use the data our app provides to efficiently schedule staff and hotel services. Also, all the users and guests of the app are stakeholders too because through their support and feedback, we can continue to improve the features and user interface. And finally, all the developers of the RUStaying app since we are investing a lot of time and effort into the creation of the system. As developers, our role is to ensure the success of the app and continue to make improvements based on the feedback from the other stakeholders.

3.2 Actors and Goals

Key
I → Initiating
P → Participating

Actors	Goals
Guests - I	Will be able to login and send requests to Hotel Staff as needed
Admins/Manager- I	Primary initiating goal is to be able to monitor customer interaction with hotel services. Will be able to log in and view requests as they need.
Hotel Staff (includes Manager) - P	Guests will be sending multiple requests for various services throughout the day. Staff receives and will act accordingly upon request. Restaurant will view orders, maintenance can view forms,etc..
Database - P	Will store all essential information from guests and general hotel information. This will be an efficient way to organize and access necessary info on guests.

3.3 Use Cases

The RUStaying app has 2 main users, the guest and admin. Each user has their own separate use cases

1. Register (guest) - to register an account on the application
2. Login (Guest & admin) - to log into a user account
3. Logout (Guest & admin) - to log out of a user account

4. Make a reservation (Guest) - to make a reservation for a room

5. Check In (Guest) - to check into the hotel

6. Room Service (Guest) - to call for room service

7. Check availability of Rooms (Admin & Guest) - to check which rooms are in use or vacant

8. Review Guest Data (Admin) - to check the information of guests, given a specific room

9. Request Status (Admin) - to be able to change or mark a guest request “in progress” or “granted”

10. Concierge (Guest) - access front desk information through the app

11. Maintenance (Guest) - to call for maintenance

12. Valet & Travel (Guest) - to call for a car to the airport or wherever the guest wants to go, or call for a valet to park the guest’s car

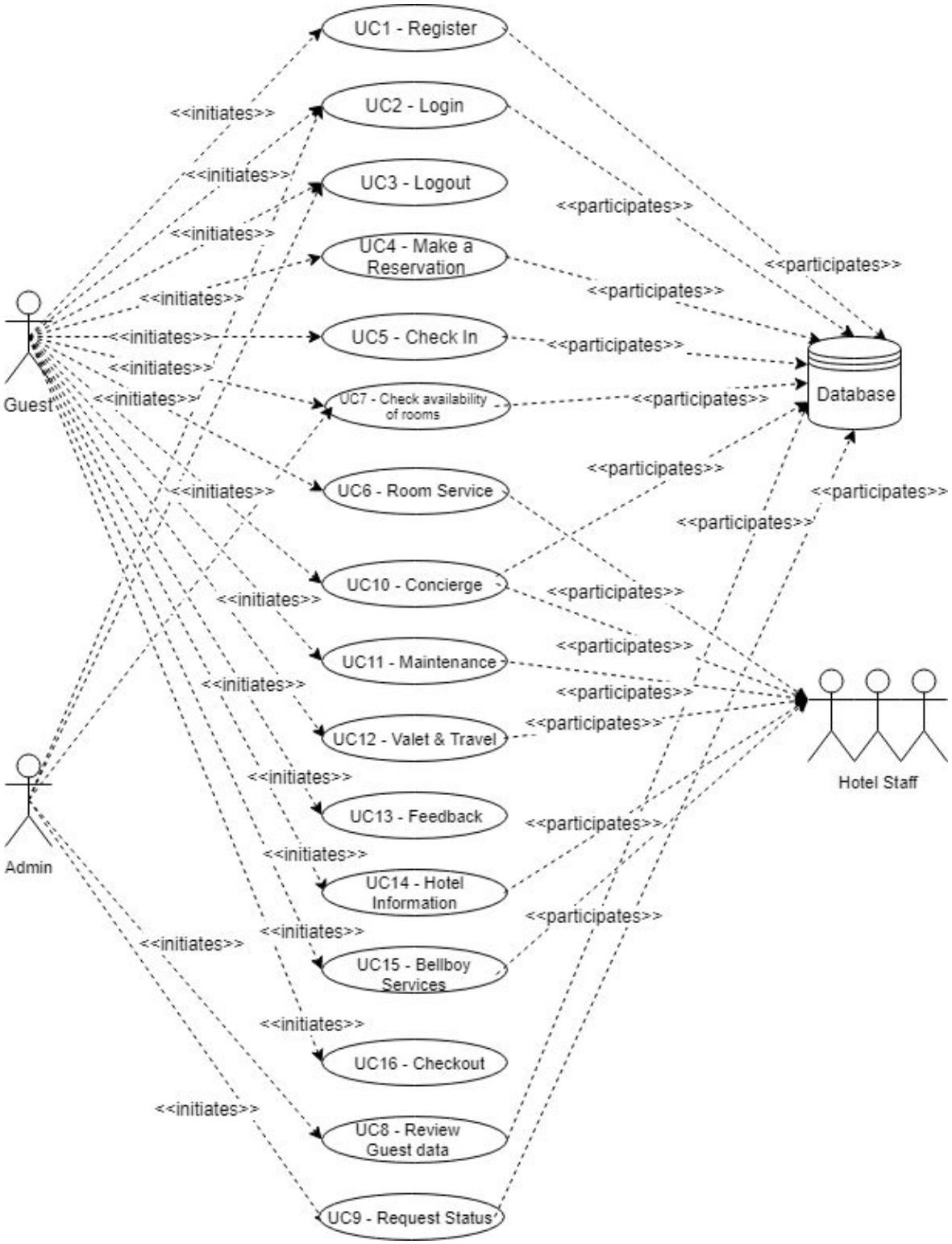
13. Feedback (Guest) - to allow the guest to provide feedback after their stay

14. Hotel Information (Guest) - to give guests a visual of the restaurant menu, gym hours, pool hours, and spa hours

15. Request Bellboy Services (Guest) - to call a bellboy to get guest’s luggage

16. Checkout (Guest) - to check out of the hotel

3.3.1 Use Case Diagram



3.3.2 Fully Dressed Use Cases

Use Case 1: Register an account
Related Requirements: REQ-1
Initiating Actors: Application User
Actor's Goal: To create an account on RUStaying app
Participating Actors: Database (keeping track of past emails used)
Preconditions: 1. The email the Guest is using to register has not been used previously in our system 2. The email and password meet certain criteria for security
Postconditions: User has an account to access the features of the app
Flow of events for main success scenarios: 1. User opens app and on login page chooses "Register an Account" 2. Fills out the required account information 3. Check for valid email and password 4. Store guest data in database 5. New account is successfully created

Use Case 2: Login to an account
Related Requirements: REQ-1, REQ-2, REQ-22
Initiating Actors: Application User
Actor's Goal: To log into an existing account created on RUStaying App
Participating Actors: Database
Preconditions: 1. User has an already existing account
Postconditions: User is able to back into his/her account
Flow of events for main success scenarios: 1. The user has opened the app and on the login page fills in their email and password 2. If email/password are incorrect, system notifies user 3. The user successfully logins into the desired account

Use Case 3: Logout
Related Requirements: REQ-1, REQ-2, REQ-3
Initiating Actors: Application User
Actor's Goal: To exit their account in the app
Participating Actors: Database
Preconditions: 1. The user already has an existing account 2. The user is currently logged into the system
Postconditions:
Flow of events for main success scenarios: 1. The user has logged in and initiates logout function 2. Database confirms user account 3. Database marks user account as logged out

Use Case 4: Make a reservation
Related Requirements: REQ-1, REQ-2, REQ-4, REQ-6, REQ-8, REQ-23
Initiating Actors: Application User/Guest
Actor's Goal: To book a room for the duration of their choosing and/or luxury of their choice
Participating Actors: Database (to check for accommodations)
Preconditions: 1. Guest must have created an account with our app 2. Guest has successfully logged in
Postconditions: Guest will receive confirmation via app
Flow of events for main success scenarios: 1. User has logged in and selected the "Make a Reservation" option 2. User fills out details of what type of room they are looking for (price, number of people, size of room, etc..) 3. System will relay information and gather available rooms and present options to user 4. User will choose an option 5. System has successfully booked the room for user

Use Case 5: Check In
Related Requirements: REQ-3, REQ-10, REQ-23
Initiating Actors: User
Actor's Goal: User would like to check in for their room upon arrival
Participating Actors: Hotel Staff, database (to check for reservation)
Preconditions: User has created an account and made a reservation
Postconditions: User received digital room key and can access services.
Flow of events for main success scenarios: 1. User has logged in and selects the "Check In" option on arrival 2. System verifies user and provides user with a digital room key.

Use Case 6: Room Service
Related Requirements:REQ-2, REQ-3, REQ-5
Initiating Actors: Application User/ Guest
Actor's Goal: To be able to call room service for a variety of purposes (Clean the room, replace toiletries, replace bed sheets, etc.)
Participating Actors: Room Staff, Maids
Preconditions: 1. Guest has been checked-in before requesting any service
Postconditions: Guest will be notified when request has been sent
Flow of events for main success scenarios: 1. User has logged in and has selected "Request Room Service" 2. System will confirm check in status of guest and return with a list of services available to guest 3. User will choose an option along with the timing of when it is needed 4. System will confirm and send notification to team responsible for service

Use Case 7: Check Availability of Rooms

Related Requirements: REQ - 6, REQ - 13
Initiating Actors: Guest, Admin
Actor's Goal: User would like to check the availability of the rooms
Participating Actors: Guest Admin
Preconditions: Users have an account created
Postconditions: App displays availability of rooms
Flow of events for main success scenarios: 1. User has logged in and selects "Make a Reservation" 2. System pulls from database availability of rooms 3. User is able to see which rooms are vacant

Use Case 8: Review Guest / Service Data
Related Requirements: REQ-13
Initiating Actors: Admin
Actor's Goal: To view the guest data, room availability and usage of hotel services
Participating Actors: Database
Preconditions: 1. Accounts created by guests 2. Guests request hotel services from app
Postconditions: 1. Display guests data (ex: Number of guests in hotel) 2. A breakdown of how much each service is used 3. List of rooms available / occupied
Flow of events for main success scenarios: *This use case is for all data storage so it <<includes>> all other use cases* 1. Guests continue to use services through the app 2. Data is kept track of and it sent to the database to store 3. The data is collected from the database frequently 4. The data is presented to the Admin in a user friendly format

Use Case 9: Request Status
Related Requirements: REQ-5, REQ-7, REQ-9, REQ-12, REQ-13, REQ-24
Initiating Actors: Admin
Actor's Goal: To be able to change or mark a guest request "in progress" or "granted"
Participating Actors: Guest, Hotel Staff, Database of Hotel Staff Activities
Preconditions: 1. Guest must have made a request for some type of service, see REQ-5, REQ-7, REQ-9, and REQ-12
Postconditions: 1. Guest request will be either labeled "in progress" or "granted"
Flow of events for main success scenarios: 1. User has logged in and makes a request from one of those offered in the app 2. System receives request and labels the request as "in progress" 3. User reports to the system when service has been completed 4. System marks requests as "granted"

Use Case 10: Concierge
Related Requirements: REQ - 17
Initiating Actors: User
Actor's Goal: To use the concierge service to obtain hotel information or other general information
Participating Actors: System
Preconditions: 1. User will type a question into the concierge tab, OR 2. Select a question from general topics listed in the tab
Postconditions: 1. User will be presented with the information they requested, OR 2. Be given the phone number of an in-person concierge at the hotel's front desk
Flow of events for main success scenarios: 1. User has logged in and selects the "Concierge" option 2. User enters a question or selects one from the general questions provided

3. System receives the information and processes it and displays an answer
    a. If the system does not have an answer to the guest's question, the phone number of the concierge at the hotel's front desk will be displayed
4. Database will return and display the answer to user

Use Case 11: Maintenance
Related Requirements: REQ-2, REQ-7
Initiating Actors: Guest
Actor's Goal: To submit a maintenance request into the app and for a hotel worker to solve the issue
Participating Actors: System, Hotel Staff, Maintenance Worker
Preconditions: 1. Guest is logged in 2. Guest needs to submit maintenance request
Postconditions: 1. Guest will have maintenance requests fulfilled 2. Confirmation message will pop up after fulfilled
Flow of events for main success scenarios: 1. User has logged in and selects "Maintenance Requests" 2. Database pulls options for Maintenance Requests 3. User will be able to select from the options presented 4. App will submit option into database 5. Database will send confirmation message that request was successfully submitted

Use Case 12: Valet & Travel Services
Related Requirements: REQ 12, REQ 25
Initiating Actors: Guest
Actor's Goal: To call for a car to the airport or wherever the guest wants to go, or call for a valet to park the guest's car
Participating Actors: Hotel Staff, Car
Preconditions:

1. Guest is logged in 2. Guest has checked in
Postconditions: 1. Guest will have their requested car waiting outside 2. Guest will be notified once the car is called and ready
Flow of events for main success scenarios: 1. User has logged in and selects “Guest Services” 2. User can select “Valet & Travel” and choose to call a car 3. Car Services will indicate that they have arrived for pick up 4. System will notify User that car has has arrived and is ready

Use Case 13: Feedback
Related Requirements: REQ-14
Initiating Actors: Guest
Actor’s Goal: To allow the user to input feedback upon check out
Participating Actors: Database
Preconditions: 1. User has checked out of the hotel
Postconditions: 1. Check out confirmation will pop up 2. Feedback form will pop up
Flow of events for main success scenarios: 1. User has logged in and selects “Check out” option at the end of their stay 2. System will output a check out confirmation 3. System will output a feedback form from Database 4. User will fill out or dismiss feedback form and submit it into the Database 5. System will output a submission confirmation

Use Case 14: Hotel Information
Related Requirements: REQ-11
Initiating Actors: Application User

Actor's Goal: To view restaurant menus, gym hours, pool hours, spa hours at the user's convenience, etc...
Participating Actors: Database
Preconditions: 1. Guest is logged in
Postconditions: 1. Guest will be informed of hotel information
Flow of events for main success scenarios: 1. User has logged in and selects "View Hotel Information" 2. System will display hotel information

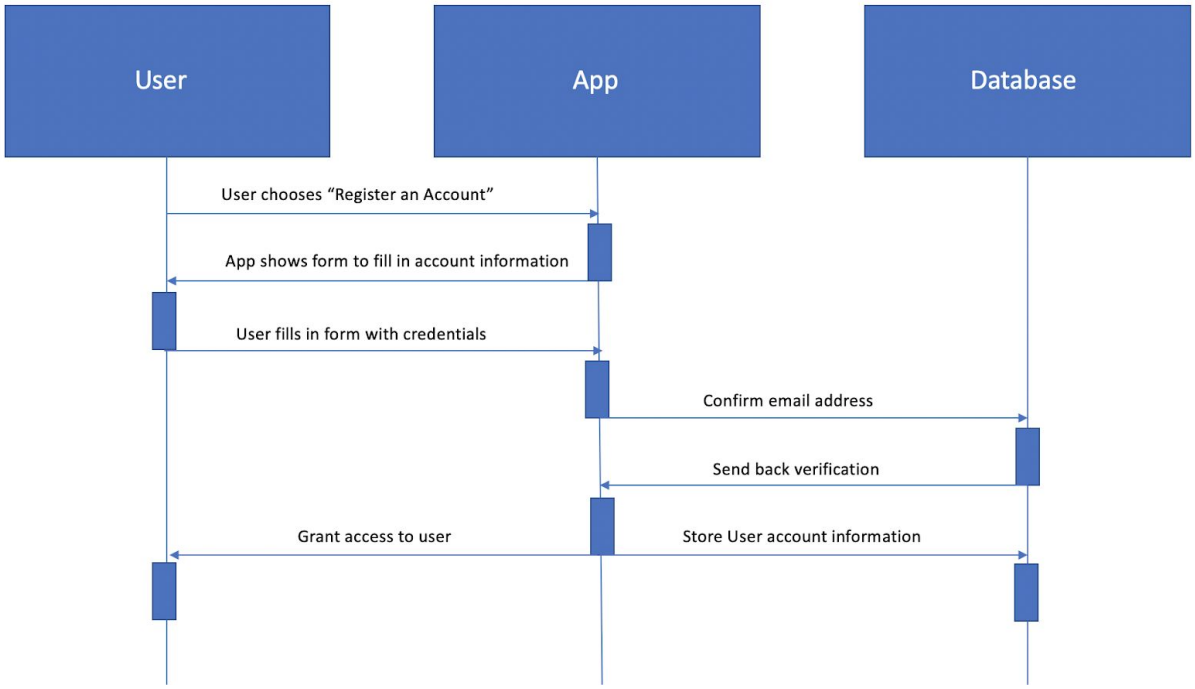
Use Case 15: Bellboy Services
Related Requirements: REQ-9 REQ-25
Initiating Actors: Guest
Actor's Goal: get their luggage moved to their room
Participating Actors: Hotel Staff
Preconditions: 1. Guest is logged in 2. Guest has checked in
Postconditions: 1. Guest will have their luggage taken up to their room 2. Guest will be notified once luggage has been taken up
Flow of events for main success scenarios: 1. User has logged in and selects "Guest Services" 2. User can select bellboy and choose to move luggage to room 3. Bellboy will indicate that they have finished moving luggage 4. System will notify User that luggage has been taken up to their room

Use Case 16: Checkout
Related Requirements: REQ-16

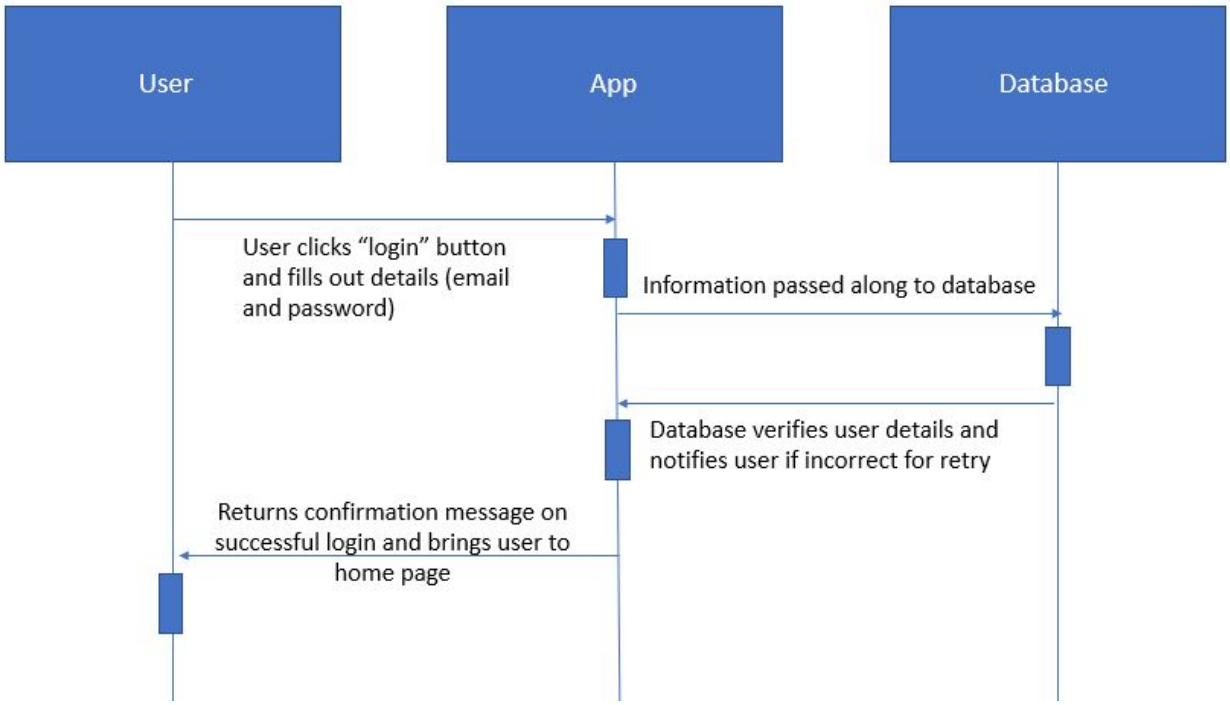
Initiating Actors: Application User
Actor's Goal: To check out of the hotel through the app
Participating Actors: Hotel Staff, Database
Preconditions: 1. Guest has finished their stay at the hotel
Postconditions: 1. Guest will be able to write a feedback report
Flow of events for main success scenarios: 1. User has logged in and selects "Check out" after finishing their stay at the hotel 2. User is given the option to have a bellboy help move luggages to the main lobby 3. Database is notified that room service is needed 4. System accesses database to update that the room is available

3.3.3 System Sequence Diagrams

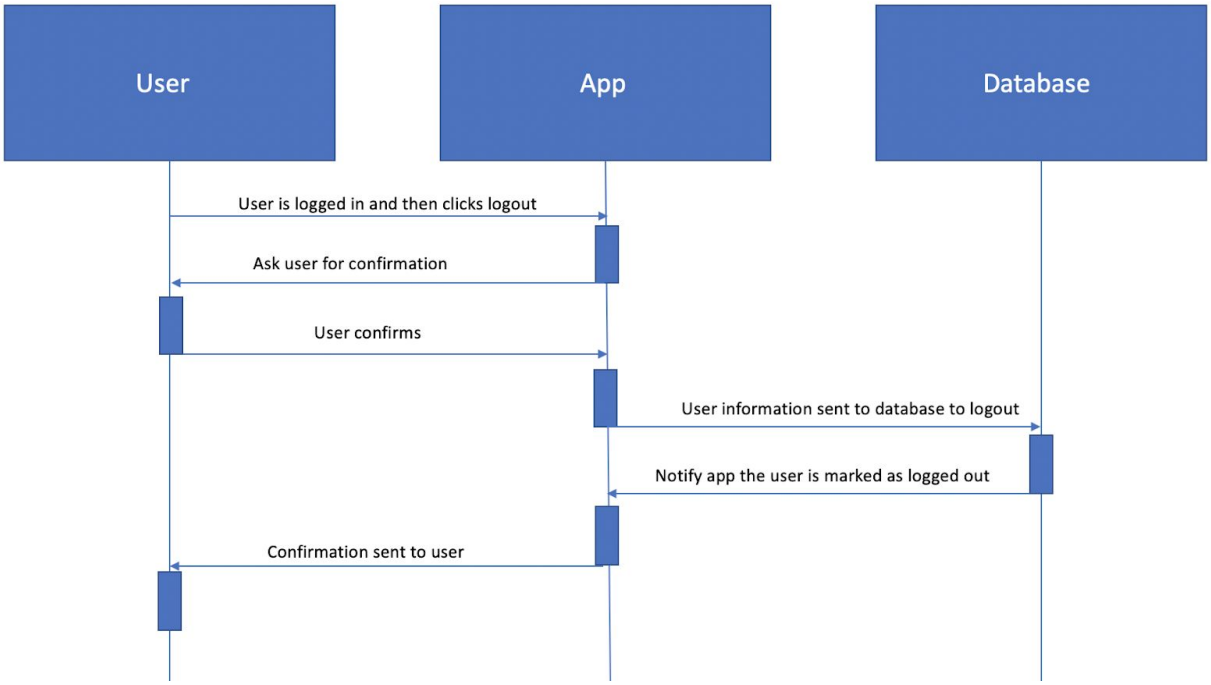
Use Case 1: Register an Account



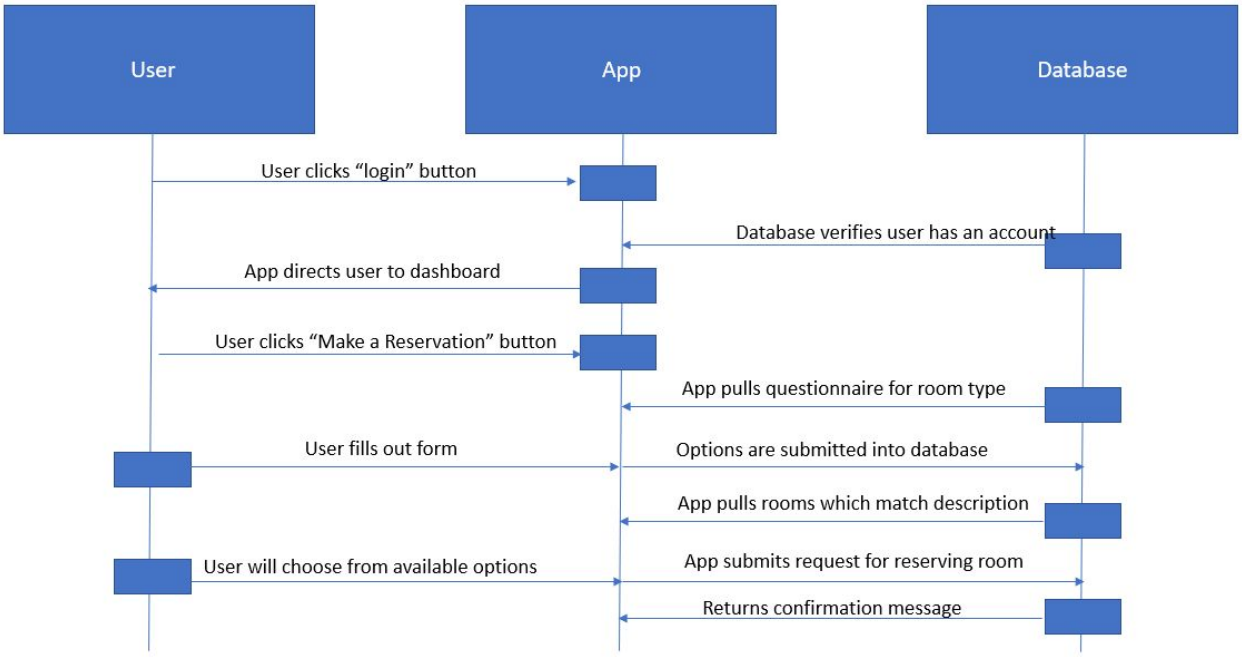
Use Case 2: Login



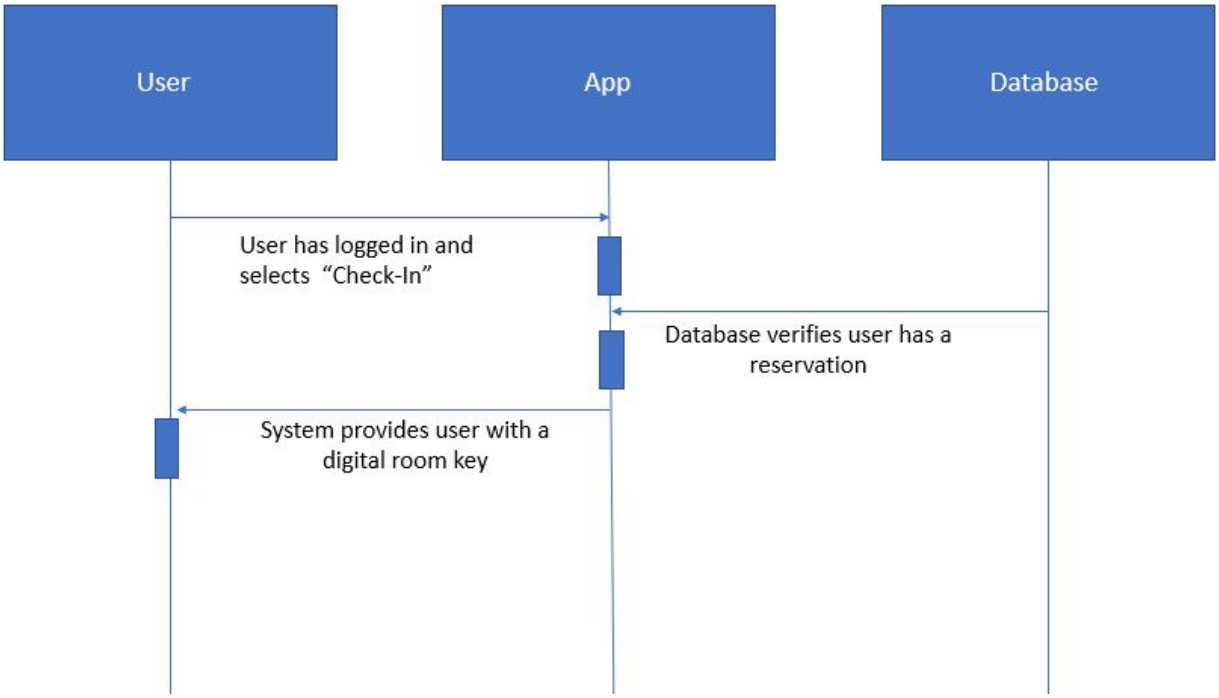
Use Case 3: Logout



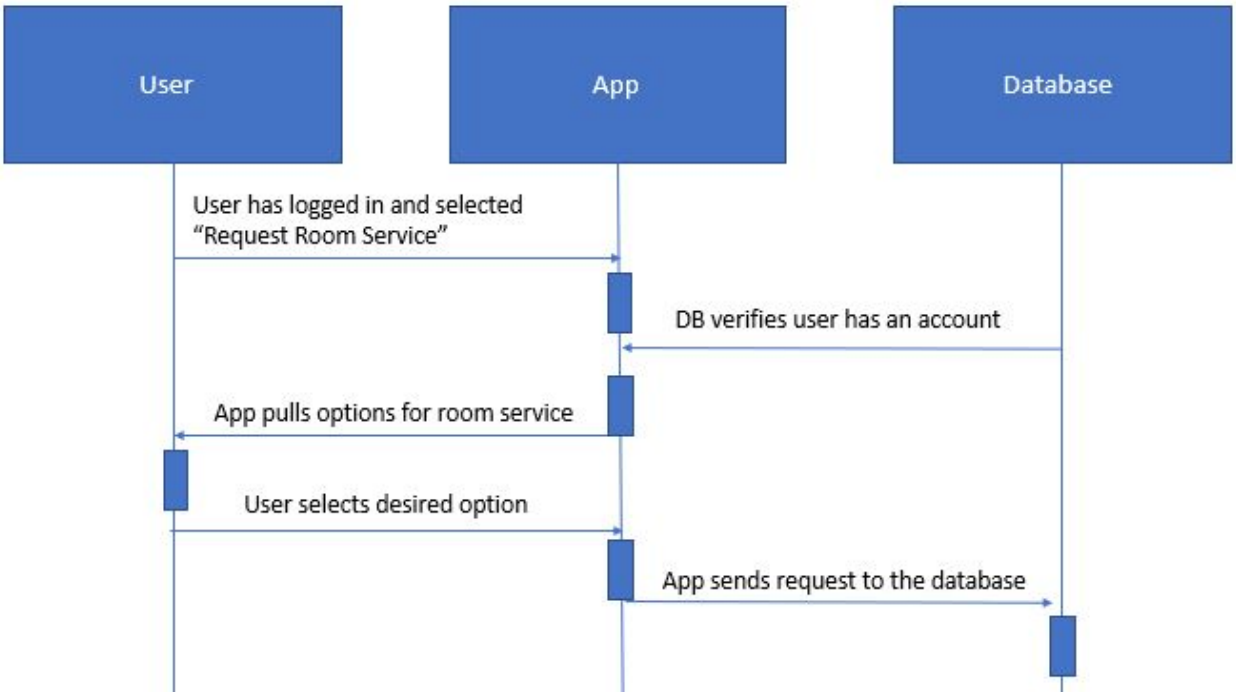
Use Case 4: Make a Reservation



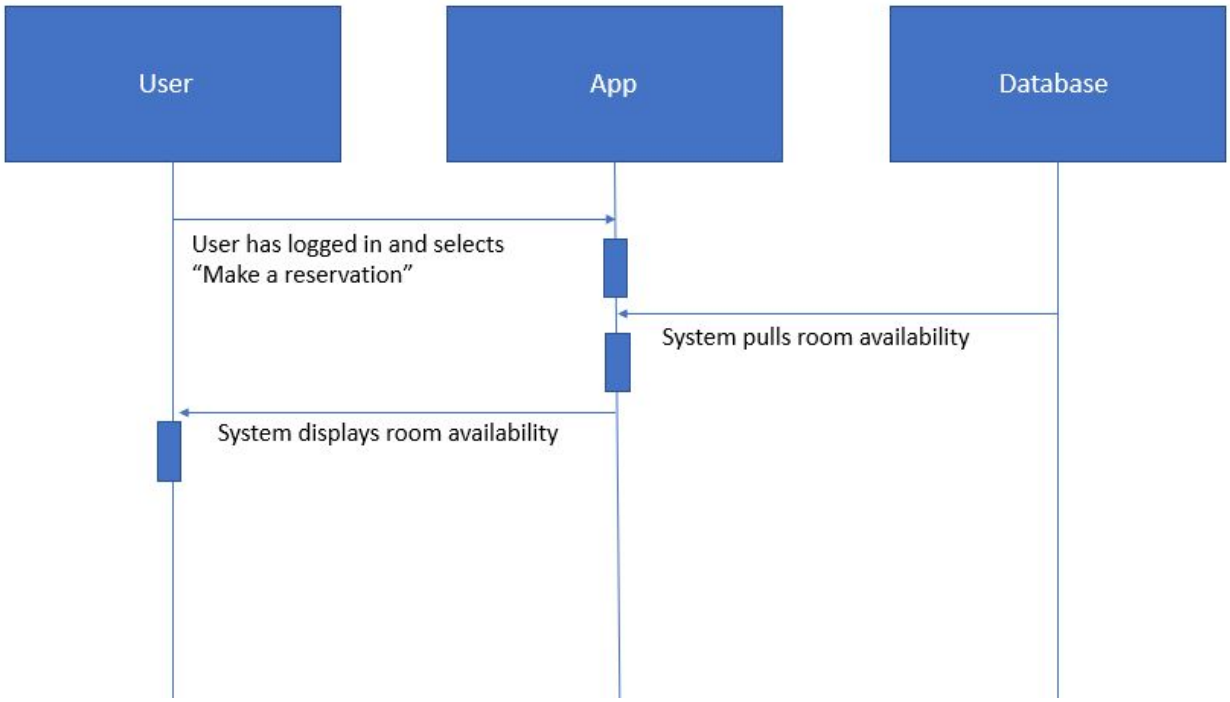
Use Case 5: Check-In



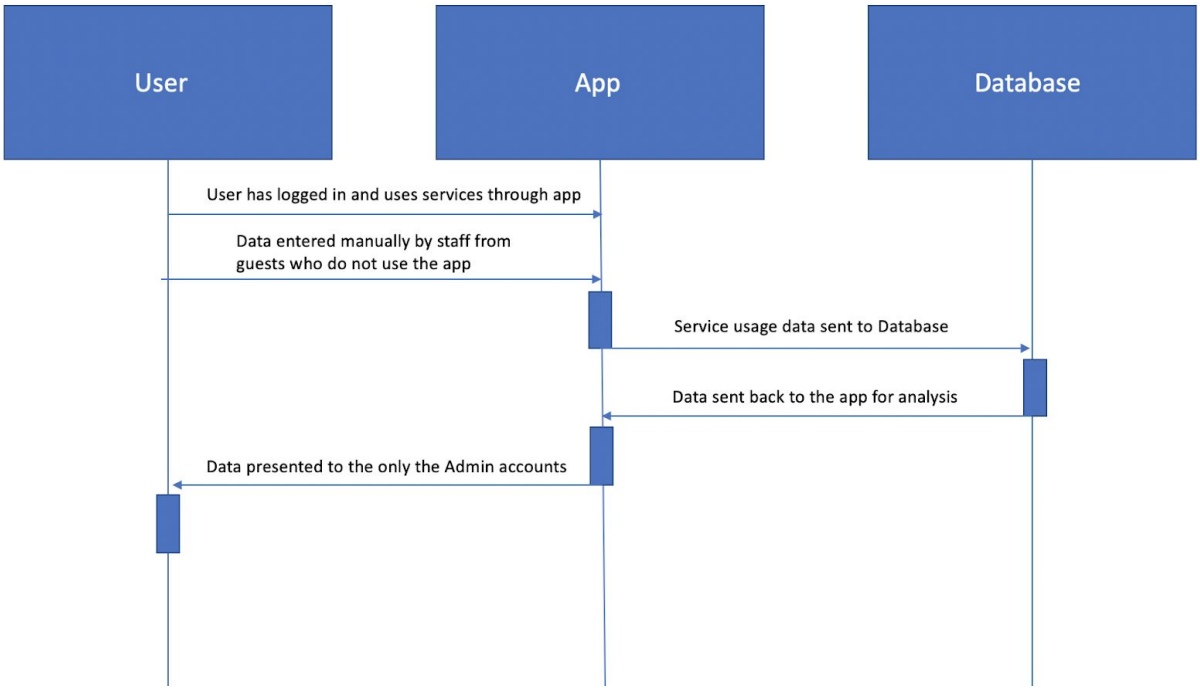
Use Case 6: Room Service



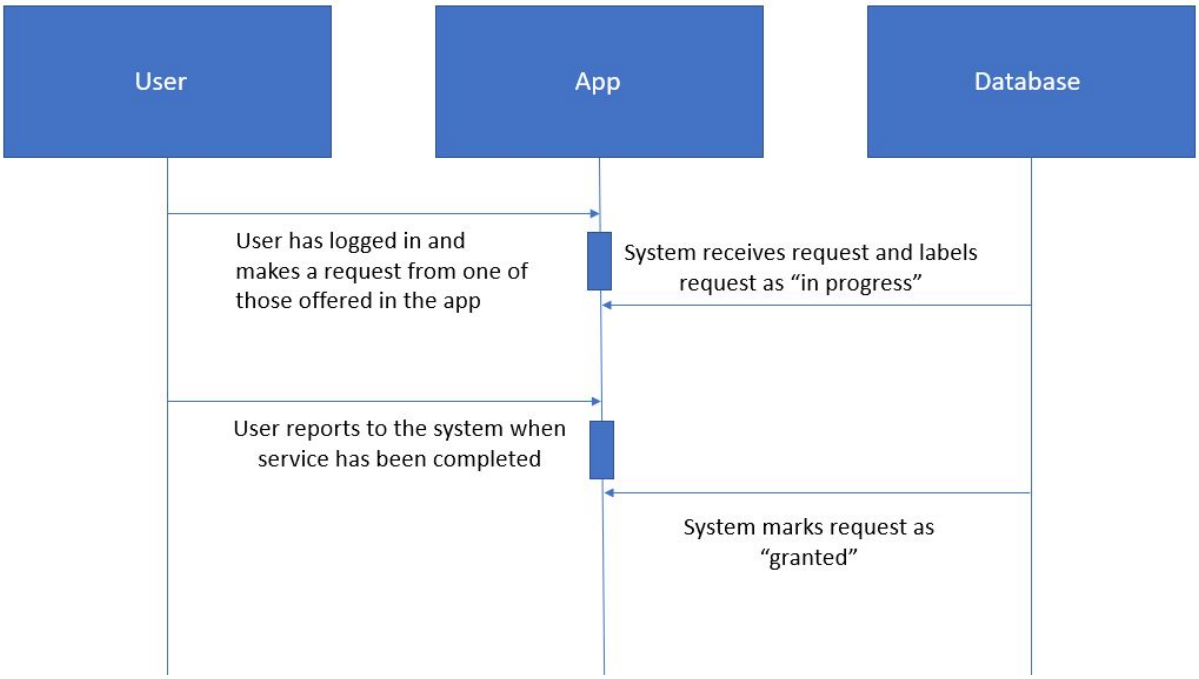
Use Case 7: Check Availability of Rooms



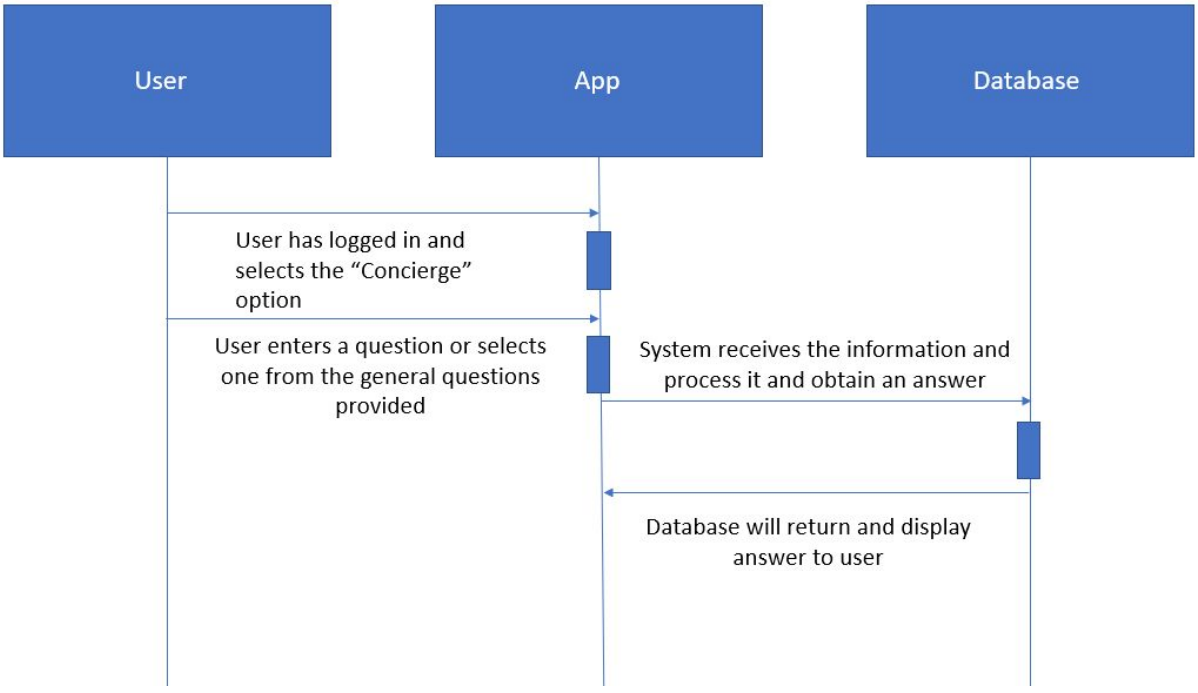
Use Case 8: Review Guest / Service Data



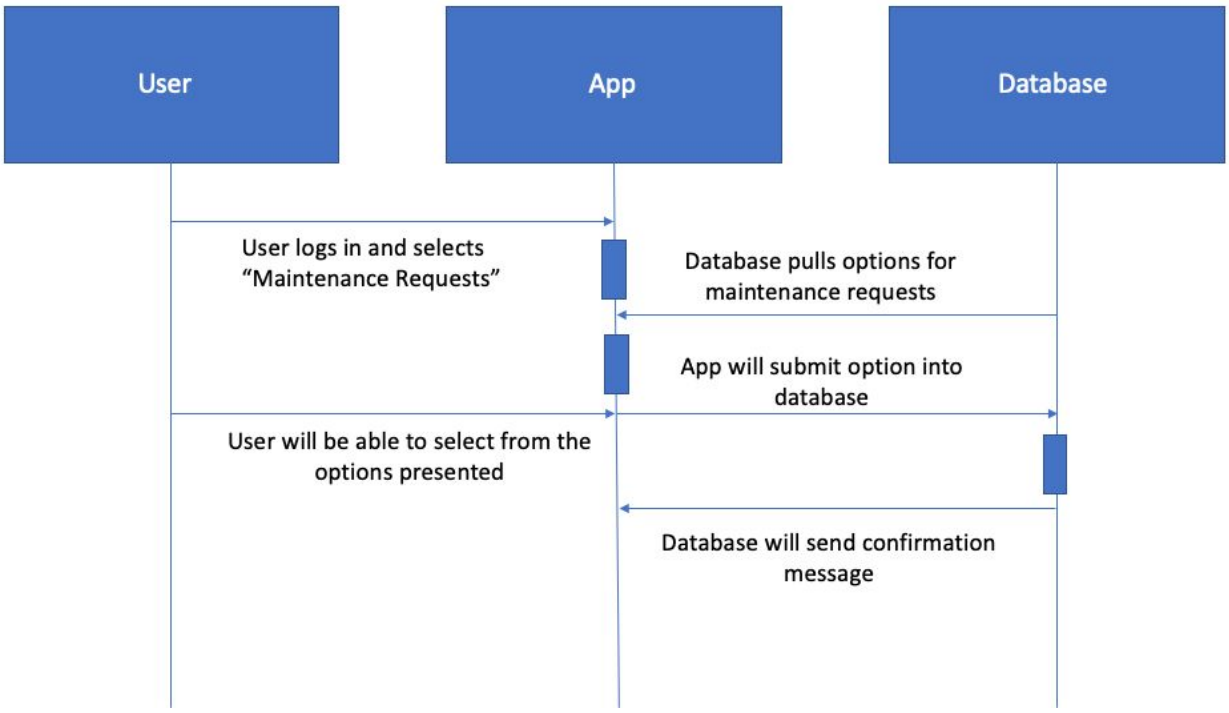
Use Case 9: Request Status



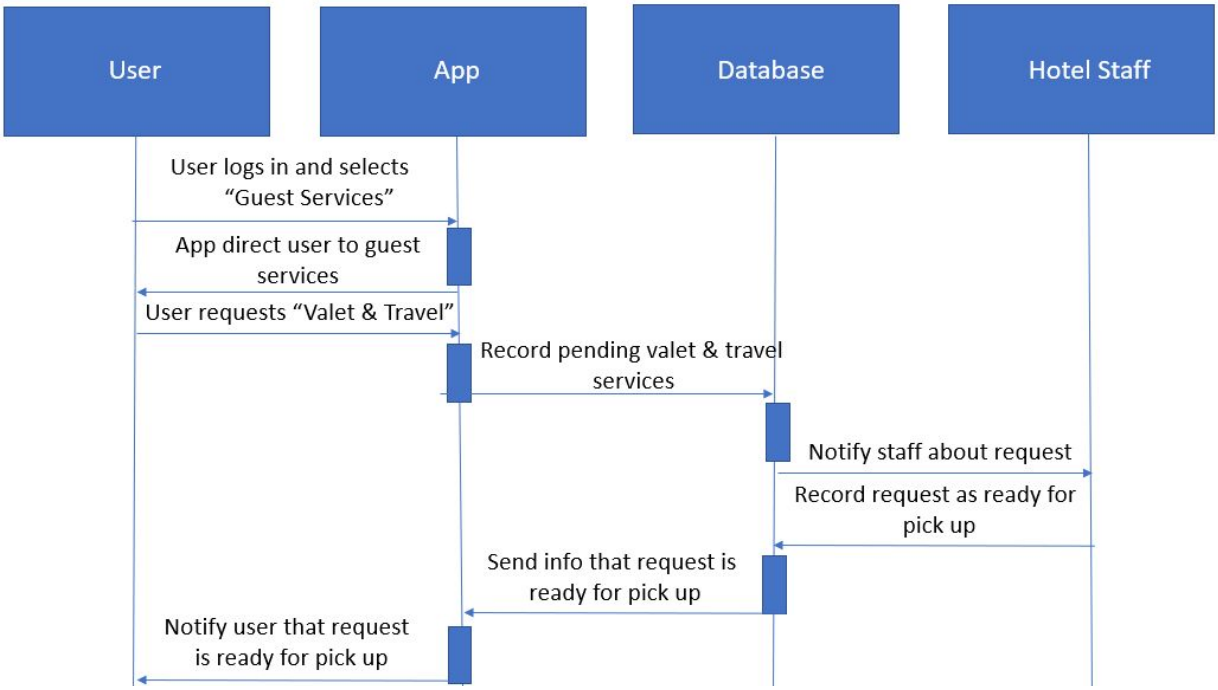
Use Case 10: Concierge



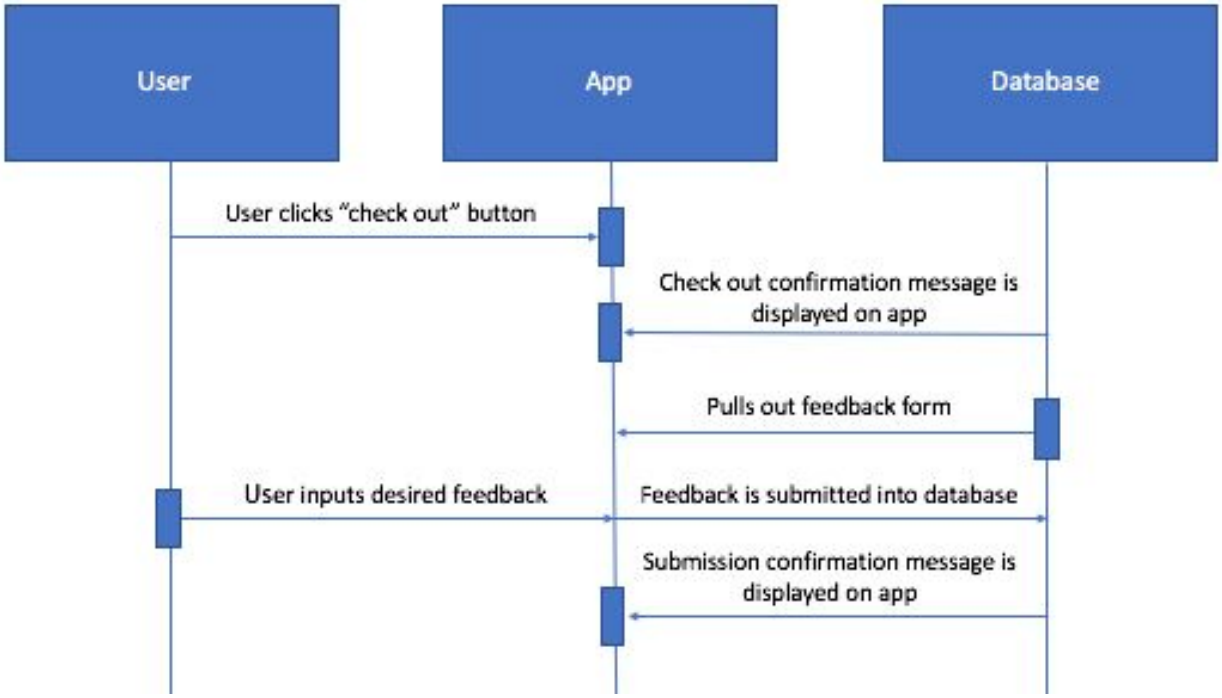
Use Case 11: Maintenance



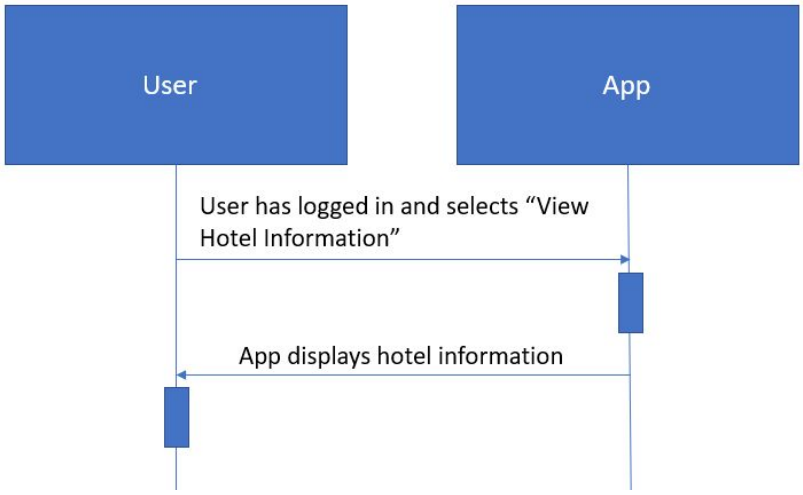
User Case 12: Valet & Travel Services



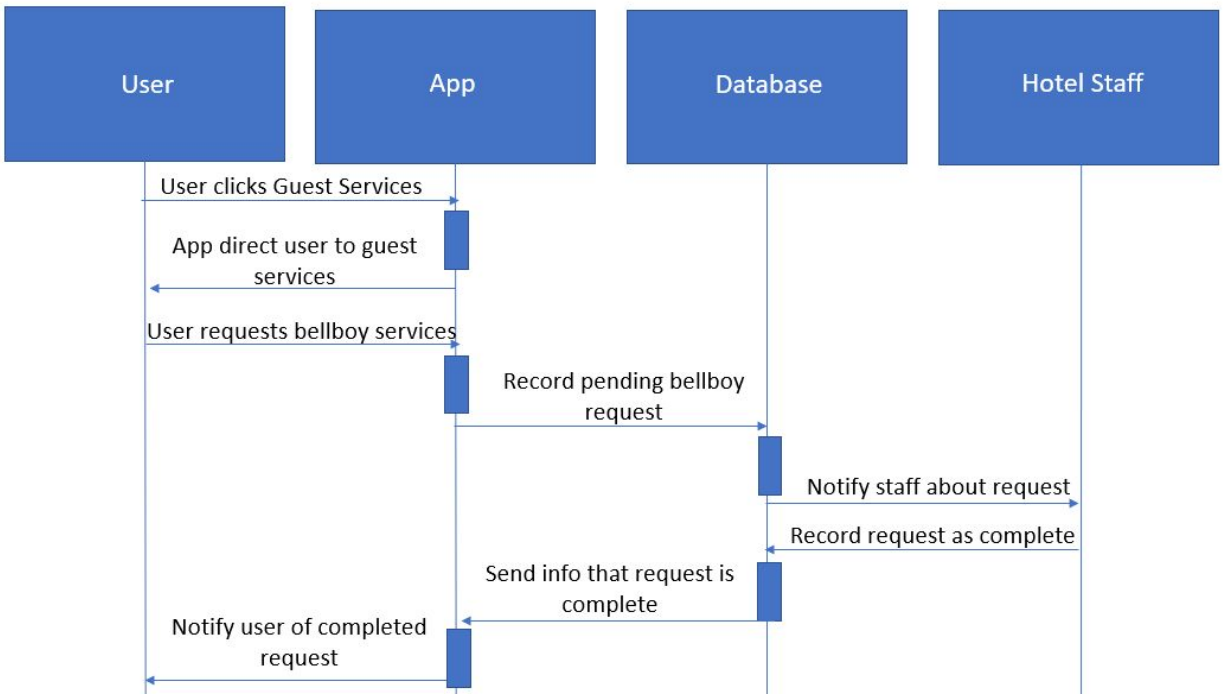
User Case 13: Feedback



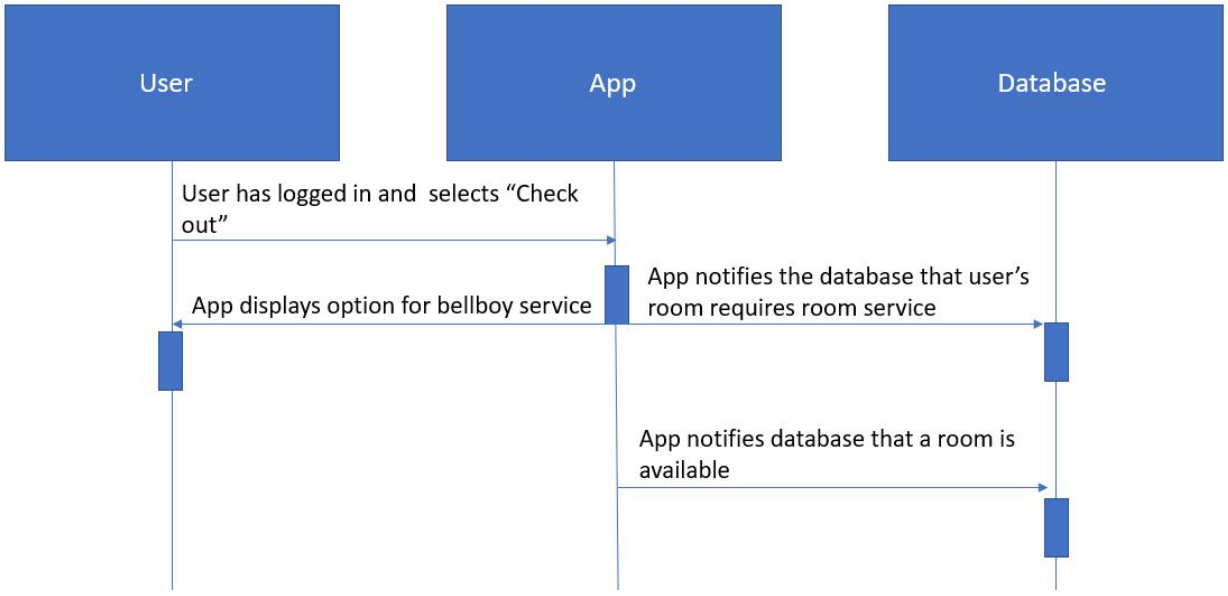
User Case 14: Hotel Information



User Case 15: Bellboy Services



User Case 16: Checkout



3.3.4 Traceability Matrix

Reqs	P.W	UC1	UC2	UC3	UC4	UC5	UC6	UC7	UC8	UC9	UC10	UC11	UC12	UC13	UC14	UC15	UC16
1	2	X	X	X	X												
2	2		X	X	X		X					X					
3	6			X		X	X										
4	6				X												
5	4						X			X							
6	8				X			X									
7	5									X		X					
8	3				X												
9	2									X						X	
10	7					X											
11	1														X		
12	3									X			X				
13	10							X	X	X							
14	1													X			
15	3																
16	6																X
17	1										X						
Max P.W.		2	2	6	8	7	6	10	10	10	1	5	3	1	1	2	6
Total P.W.		2	4	10	21	13	12	18	10	24	1	7	3	1	1	2	6

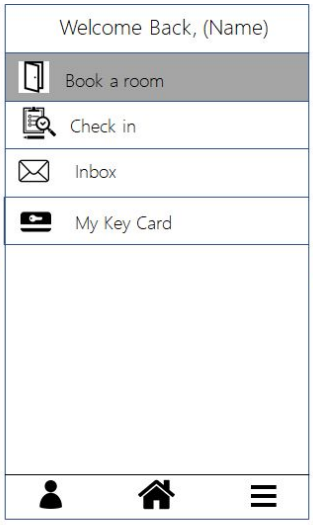
Chapter 4

User Interface Specification

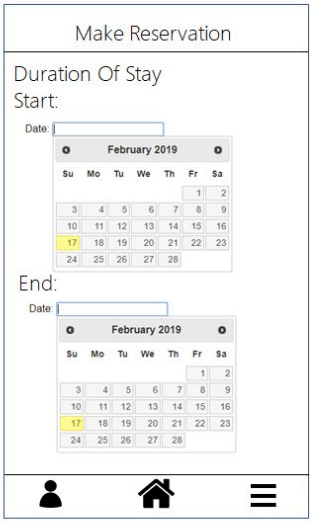
4.1 Preliminary Design

Use Case 4: Make Reservation

Start page where user can login

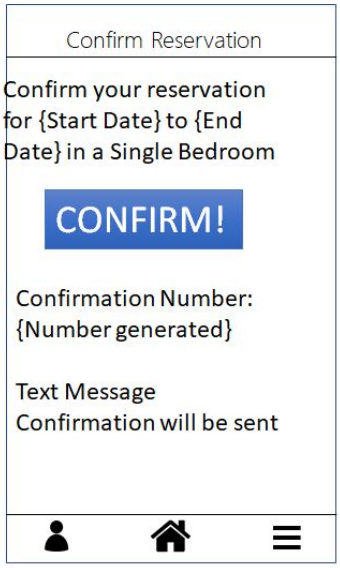


User selects Book a Room



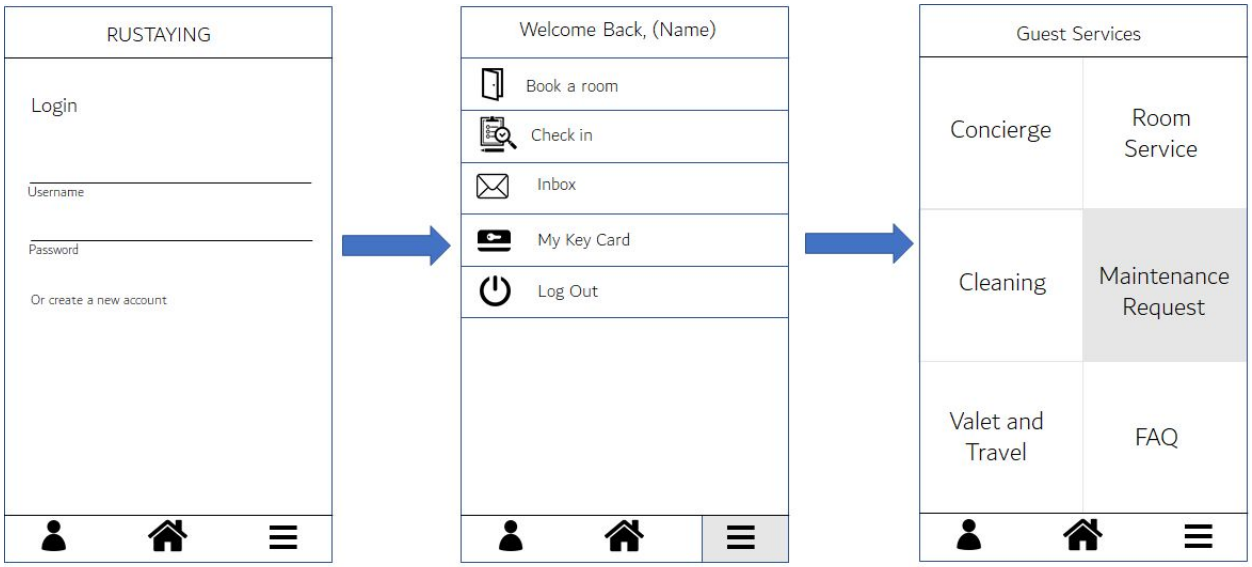
User selects start and end date

User selects from available rooms

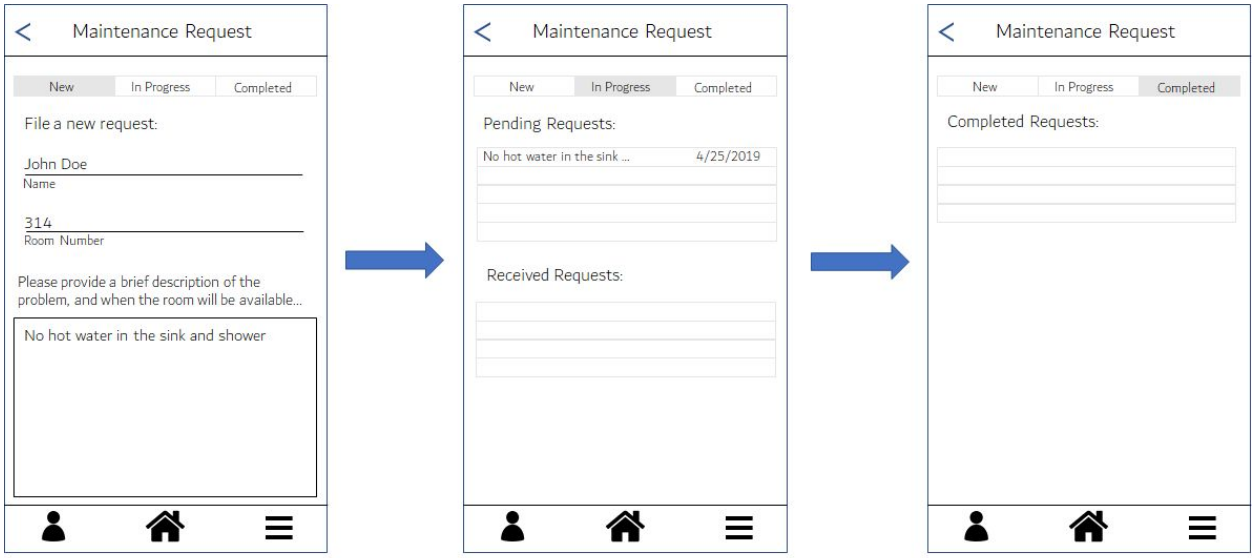


User confirms booking

Use Case 11: Maintenance Request



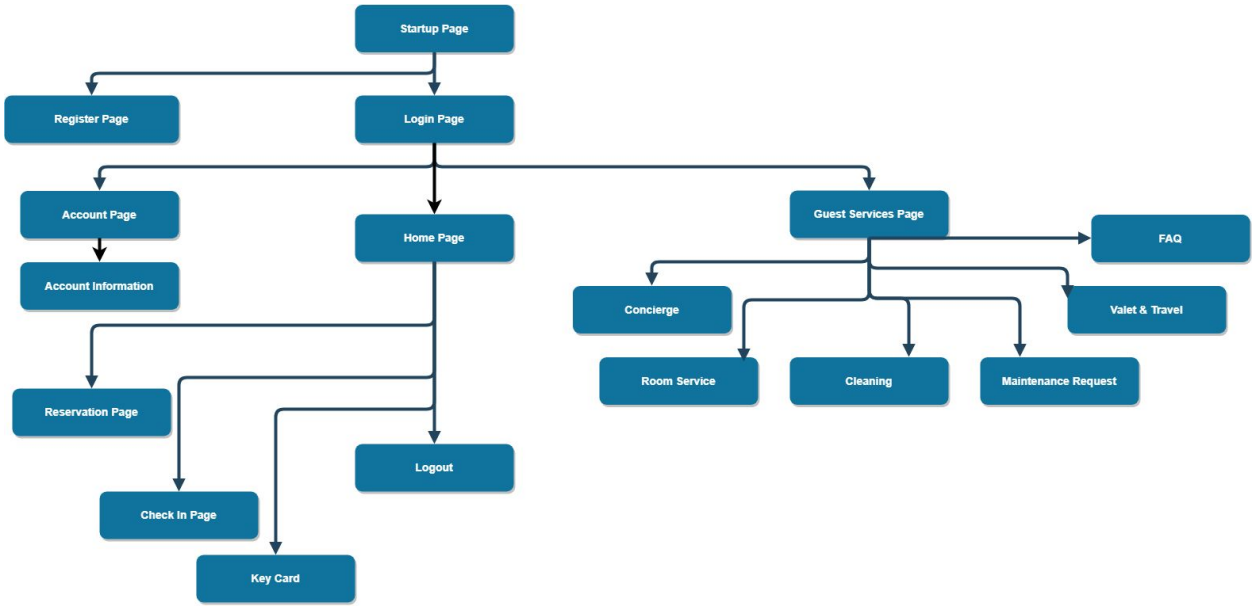
User logs in | Select the Guest Services | Select Maintenance Request



New Requests | Pending/In progress Requests | Completed Requests

4.2 User Effort Estimation

4.2.1 Navigation Tree



Since we are building an android app navigation through each page is one click that will take you to a new page in the app. So from the guest services page to access each of the different features it is one button click. Each page also provides a back button to return to the previous one.
**Guest services page may have additional features in the future

Chapter 5

Domain Analysis

5.1 Domain Model

The domain model is a conceptual model of a domain that incorporates both behavior and data. This model can be used to solve problems related to that domain. By using the requirements listed from each use case, the domain model shows exactly how the user will interact with the system and how the system will complete the requirement. It is helpful to map out the interactions from the user and also the interactions within the system itself.

5.1.1 Concept Definitions and Responsibilities

Responsibility Description	**Type**	**Concept Name**
RS1. Main source for all other subsystems to interact with. Coordinate actions and delegate work based on user interactions.	D	Controller
RS2. Verify if user credentials are valid	D	UserVerification
RS3. Create a new guest account	K	UserManagement
RS5. Store user account information (username, password, email, previous reservations etc…)	D	UserManagement
RS4. To allow user to make a reservation	D	RoomControl
RS6. To allow the guest to check-in for their hotel reservation	K	RoomControl
RS7. Guest can request room service	D	HotelServices
RS8. Guest can make maintenance requests for their room during their stay	D	HotelServices
RS9. Guest can request a bellboy for luggage	K	HotelServices
RS10. Guest can use the hotel car service	K	HotelServices

RS11. System will keep track of available / unavailable rooms	D	RoomTracking
RS12. System will keep track of each hotel service as it is requested	D	HotelTracking
RS13. Review all data on services used by guests	D	AdminControl
RS14. Present data in viewable manner to analysis and predictions	D	AdminControl

Some concepts interact directly with the database but serve different functions so they are listed as separate concepts. The concepts that directly interact with the database are: UserManagement, RoomTracking, HotelTracking and AdminControl. The other concepts are control concepts that the user can interact with through the controller to achieve functionality.

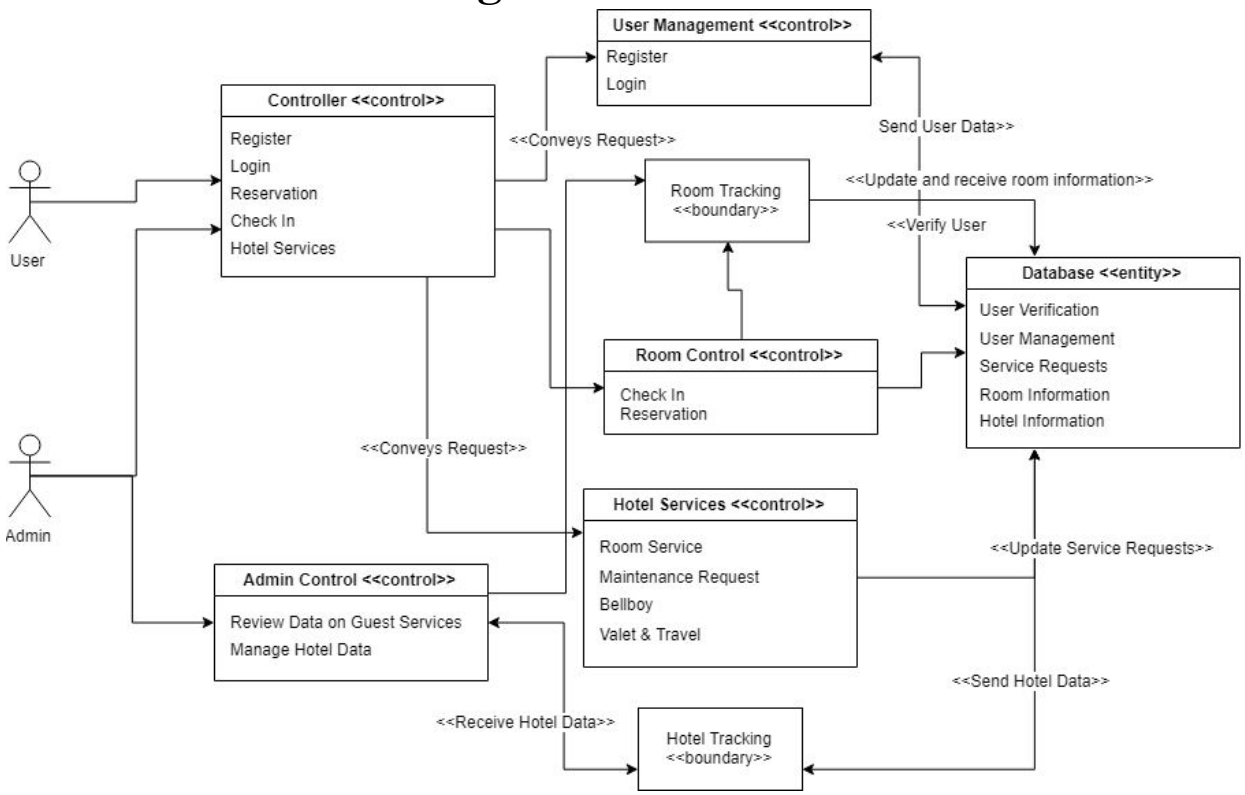
5.2 Concept Associations

Concept Pair	Association Description	Association Name
UserVerification ↔ UserManagement	User credentials sent to UserManagement to check if it is valid and receives back a confirmation	Validate User
Controller ↔ RoomControl	As user interacts with app, the controller will send room reservations and room check-in information to RoomControl concept	Room Bookings
RoomControl ↔ RoomTracking	As RoomControl is used, the information is passed to RoomTracking, which will store the data and also allow for analysis	Room Analysis
Controller ↔ HotelServices	When users request any sort of service available through our app, the task will be delegated to HotelServices concept, which will appropriately handle the request	Handle Service Request
HotelServices ↔ HotelTracking	As HotelServices handles the guest requests, the data will be sent to HotelTracking, which will store the data and allow for analysis	Hotel Analysis
RoomTracking ↔ AdminControl	Once room data is analyzed in RoomTracking concept, it will be sent to AdminControl, where the admin can easily view it	Admin Room Data
HotelTracking ↔ AdminControl	Once hotel services data is analyzed in HotelTracking, it is sent over to AdminControl and presented so that the admin can easily notice trends and patterns	Admin Hotel Data

5.3 Attribute Definitions

Concept	Attributes	Attribute Definitions
UserVerification	User's account data	Contains user's identification including name, email, phone number, and home address associated with the account
UserManagement	Store account data	Stores newly made account or recently edited account information
RoomControl	Room data	Shows available rooms that are reservable
	Check In	Allows user to check in
HotelServices	Request data	Directs user requests to corresponding service
RoomTracking	All room data	Contains data on available and unavailable rooms
HotelTracking	All Request data	Contains data from all user requests in the hotel
AdminControl	Hotel data	Contains information on all in progress or completed hotel services

5. 4 Domain Model Diagram



The domain model was derived by looking at our concept definitions and which use cases they fulfilled. The main component in our domain model is the controller that connects to all other concepts in our device so it fulfills all our use cases. From the controller the user can access the other concepts in our application. The main ones being user management, room control, and hotel services. From user management the user can register and log in to the app to access the different components of our app. The next one room control fulfills UC-4 and UC-5. The hotel services controller gives users access to the various features described in our use cases that a user can do. The database serves as the entity that stores all this information about user, hotel information, room information, requests, and fulfills requests to send and store data. Finally the admin control allows the admin of the hotel to have additional features to fulfill our use cases. With boundaries such as room tracking and hotel tracking that communicate with the database the admin control allows the admin to have access to all hotel data sent in by guests and retrieve it to perform future predictions about management and efficiency.

5.5 Traceability Matrix for Domain Concepts

PW	User Cases	Controller	UserVerification	UserManagement	RoomControl	HotelServices	RoomTracking	HotelTracking	AdminControl
2	UC1	X	X	X					
4	UC2	X	X						X
10	UC3	X							X
21	UC4	X	X	X	X				
13	UC5	X	X		X				
12	UC6	X				X		X	
18	UC7	X					X		X
10	UC8	X		X					X
24	UC9	X						X	X
1	UC10	X							
7	UC11	X				X		X	
3	UC12	X				X		X	
1	UC13	X		X					
1	UC14	X				X			
2	UC15	X				X		X	
6	UC16	X							

5.6 System Operation Contracts

UC-1 - Register an account

- Preconditions:
    - Email guest is using to register has not been previously used
    - Password meets security standards
- Postconditions:
    - User now has an account that can be used to login into the application

UC-2 - Login to account

- Preconditions:
    - User has registered an account
- Postconditions:
    - User can enter email and password to login

UC-3 - Logout

- Preconditions:
    - User has registered an account
    - User is currently logged into the application
- Postconditions:
    - A different user can now log in or register an account

UC-4 - Make a reservation

- Preconditions:
    - Guest must have an account be logged into the application
- Postconditions:
    - User will receive room options
    - Guest will receive an email confirmation of their reservation

UC-5 - Check-in

- Preconditions:
    - Guest has made a reservation
    - Guest has arrived at the hotel
- Postconditions:
    - Guest can access service of the hotel they are currently staying at

UC-6 - Room Service

- Preconditions:
    - Guest has checked into their hotel
- Postconditions:
    - Guest will have their request granted
    - Guest will be notified once their Room service request has been complete

UC-7 - Check Availabilities of Rooms

- Preconditions:
    - User must be logged into the application
- Postconditions:
    - Application will display what rooms are occupied and what rooms are not

UC-8 - Review Guest/Service Data

- Preconditions:
    - Have an Admin account
    - Have Guests staying at your hotel
- Postconditions:
    - Display Guest Data (ex: number of Guests, their rooms, etc.)
    - List of Requested Services per Guest
    - List of Rooms and their occupants

UC-9 - Request Status

- Preconditions:
    - Guest must have made a request for some type of service
- Postconditions:
    - Request will be labeled “in-progress”, “granted”, or “denied”

UC-10 - Concierge

- Preconditions:
    - Guest can type a question to the concierge through the concierge tab in tha application
    - Guest can also select question from general topics listed in the
- Postconditions:
    - Guest will be given the answer for their question
    - Guest may also be given a phone number of in-person concierge at front desk

UC-11 - Maintenance

- Preconditions:
    - Guest is logged in and is on the tab to submit maintenance requests
- Postconditions:
    - Guest will have maintenance request fullfilled
    - Guest will receive confirmation message once maintenance request is fulfilled

UC-12 - Valet & Travel Services

- Preconditions:
    - Guest is logged into the application
- Postconditions:
    - Guest will have their car, driven out or into the garage
    - Guest can arrange transportation and will be notified once it is ready

UC-13 - Feedback

- Preconditions:
    - Guest has checked out of hotel
- Postconditions:
    - Guest will be prompted with a feedback form

UC-14 - Hotel Information

- Preconditions:
    - Guest is logged in
- Postconditions:
    - Guest will be able to view a hotel's information. (e.g. Restaurants, Facilities, Hours)

UC-15 - Bellboy Services

- Preconditions:
    - Guest is checked in
- Postconditions:
    - Guest will have luggage taken to their room
    - Guest will be notified once luggage has been taken up

UC-16 - Checkout

- Preconditions:
    - Guest has finished their stay at the hotel
- Postconditions:
    - Guest will leave hotel

Guest will be sent feedback form

Chapter 6

Plan of Work

The plan of work is to describe our course of action as a group after submitting Report 1. It consists of milestones with projected due dates to complete the software in time.

These milestones will be ranked either 1 or 2 based on if it high priority or low priority respectively. High priority milestones are ones that need to be accomplished by projected due dates in order to meet deadlines and are needed for milestones that come after. On the other hand, low priority milestones are not crucial to the project timeline but should be worked on regardless.

It should be noted that our short term milestones are more detailed than our long term milestones because things are subject to change down the line as we progress through development.

6.1 Milestone 1 - Working front-end modules for most-used pages

Rank: 1
Projected due date: 02/28/2019
Description: To have the app do the bare minimum, we have to finalize the app display so that we have a template that we can work with and maintain consistency throughout the project
The most-used pages are accordingly:

- Sign-up page
- Login page (Not logged in)
- Home page (Logged in, user accesses functions through here)
- “Make a reservation” page

6.2 Milestone 2 - Implement the combination of most-used pages in an app framework

Rank:1
Projected due date: 03/15/2019
Description: For the most used pages(Milestone 1), the the functionality will come later down the line. Laying down this framework does not require databases but rather a working app where we can jump from one page to another seamlessly after login. This helps build our foundation to build the rest of the project on.
Combining the most-used pages in app form:

- Sign up page
- Login page (Not logged in)
- Landing page (Logged in, user accesses functions through here)

- “Make a reservation” page

6.3 Milestone 3 - Implement functionality for most-used pages

Rank:1
Projected due date: 03/15/2019
Description: To have a working app which has the capability of the most basic utility, the most-used pages need to be in performing condition. Space and efficiency is not a priority in this situation but a mock customer should be able to :

- Sign up for an account
- Log in to the app
- Make a reservation with specific dates
- Receive an email for confirmation
- Log out successfully

6.4 Milestone 4 - Implement a managerial login which has more security clearances than regular guest login

Rank:1
Projected due date: 03/20/2019
Description: Manager should be able to login and view customer interaction by selecting a customer in database and pull tables of info from it. Primarily for hotel security and app maintenance as well to make sure no requests go unnoticed/unfulfilled.

6.5 Milestone 5 - Design a page for remote check-in/out

Rank: 2
Projected due date: 03/24/2019
Description: In order to implement remote functionality for guest’s convenience, we need to first create a display template for the users to choose between the different options. These options are included below.

- Check-in
- Check-out
- Digital Key

6.6 Milestone 6 - Implement remote functionality for guest’s convenience

Rank: 1
Projected due date: 03/24/2019
Description: At the guest’s convenience, they can check in/out and request a digital key if they would like to help make their stay more comfortable. Check in/check out application will send a request to the appropriate staff who will be ready for a guest checking in/ dropping off key. Digital key will generate a key in the backend, store it in the database and send a copy to guest.

6.7 Milestone 7 - Design pages for guest services

Rank: 2
Projected due date: 03/31/2019
Description: After guest has checked in, they should be able to utilize services that the hotel provides. Displays without functionality will be needed before we link to the app and start backend. Included as below.

- Room Service
- Maintenance Request
- Bellboy Service
- Driver/Taxi request

6.8 Milestone 8 - Implement guest services functionality

Rank: 1
Projected due date: 03/31/2019
Description: Since each service available is unique and has its own responsibilities, we have to implement a backend system which will take the data received in the app, cross check with database tables , and output accordingly.

6.9 Milestone 9 - Design pages for static FAQ

Rank: 2
Projected due date: 04/07/2019
Description: A display page is needed to showcase answers to frequently asked questions by guests and to have available a number to reach the front desk

- Concierge Service
    - FAQ
    - Phone number to call front desk
- Hotel information
    - Restaurant menu
    - Gym hours
    - Pool/Spa hours

6.10 Milestone 10 - Implement forms for requests and feedback

Rank: 2
Projected due date: 04/07/2019
Description: Static pages which have the same functionality of google forms where customers can send feedback and/or ask more questions if they so choose.

6.11 Milestone 11 - Debug and begin prepping final demo presentation

Rank: 2
Projected due date: 04/15/2019

Description: Apps will always have some mistakes which can only be identified during runtime. Our team will thoroughly test and document all possible cases of normal usage and solve any apparent errors. We will also be making a routine that the average guest would go through during their stay, this also helps with extra testing.

6.12 Milestone 12 - Final Check

Rank: 1
Projected due date: 4/20/2019
Description: Our team will perform a final check on all our features and functionality of the app. We will run all possible tests to make sure that the app is 100% functional and has no problems. This milestone will be the launch of our app.

6.13 Project Ownership & Breakdown

App User Interface

- Simple interface for guests and admins to use and access the app
- Design implementation allowing users to access all the services of the hotel
- Send notifications of check in, check out, and key-pass information

Database Server

- We will be using SQLite for our database server
- The database will be used to store and verify guest/admin account information
- Also we will store all the data collected from users and present it for the admin UI

Below is a detailed breakdown of the various parts of the project and how the work is split up. We defined the subgroups and which parts each team is going to develop. This breakdown is subject to change as we progress through development.

Subgroups based on feature modules:

Features	Subgroup
Check In Module (Guest UI focused)	Zain, Mohammed, Rameen
Guest Services Module (Guest UI focused)	Purna, Mandy, Shilp
Additional Amenities/Administrative Services Module (Admin UI Focused)	Mathew, Eric, Thomas, Keya

Specific Tasks Assigned to Members

User Login and Account Creation

- **Rameen** will develop the main interface to allow guests and admins to create accounts and login with their email and password. **REQ-1 & REQ-2**.

Reservation Management

- **Zain** and **Mohammed** will develop the infrastructure required for users to make reservations and check in to their room upon arrival at the hotel. **REQ-3 & REQ-4**

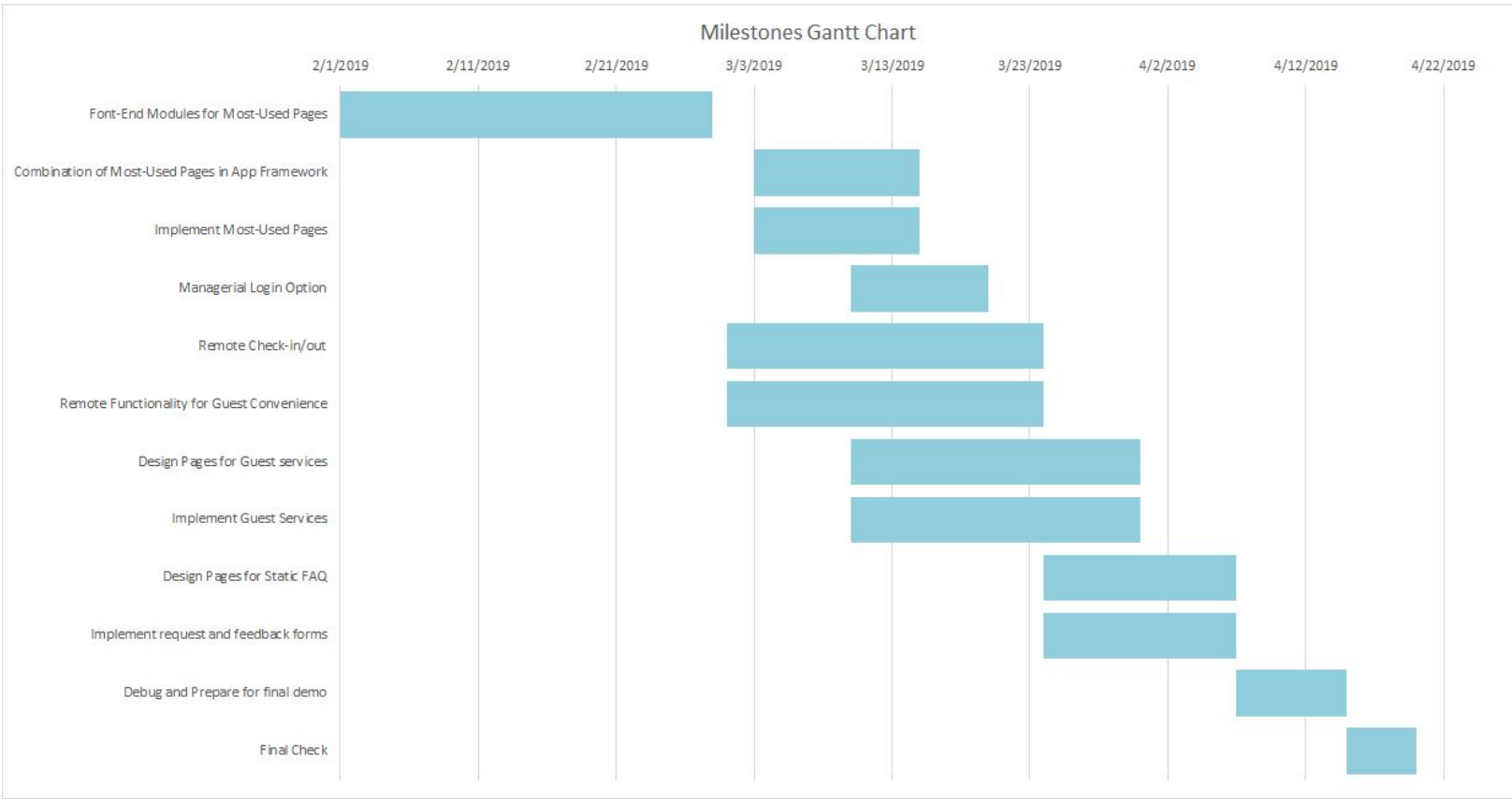
Guest Services

- **Purna** will develop the room service and car service features. This will be implemented within the UI and gives the user an estimate of how long the service will take to complete based on hotel staff available. **REQ-4 & REQ-12**
- **Mandy** will develop the digit room key feature and the bellboy service. For the room key feature, there will be a simple security aspect so that guests can only access their rooms. **REQ-9 & REQ-10**
- **Shilp** will develop the maintenance request and feedback features. Both of these features will prompt the guest to enter specific information which is sent back to the administrative side of the app for review. **REQ-7 & REQ-14**
- Additionally, our subgroup together will implement a simple feature to communicate with hotel staff directly. **REQ-15**

Additional Amenities/Administrative Services Module

- **Mathew** will implement the feature to keep track of available and unavailable rooms in the hotel, based on current guests and future reservations made. **REQ-6**
- **Eric** will implement a simple security system for storing passwords and account information safely. Also develop a simple feature in the UI that lists additional hotel information and frequently asked questions. **REQ-3 & REQ-11**
- **Thomas** and **Keya** will keep track of all other hotel services usage. This includes keeping track of how often each of the services is used and the peak times where more staff is needed. This is a difficult task because it involves data tracking and analysis. **REQ-13**

6.14 Gantt Chart



References

New Technologies Will Revolutionize the Hospitality Industry
https://www.forbes.com/sites/forbestechcouncil/2018/06/28/new-technologies-will-revolutionize-the-hospitality-industry/#51d50bc673c3

How Hotel Automation Benefits Guests
https://www.benbria.com/automation-benefits-guests-hotels/

7 Ways Hotels Are Moving to an Automated Future
https://hospitalitytech.com/7-ways-hotels-are-moving-automated-future

Alice App
https://itunes.apple.com/us/app/alice-life-improving-concierge/id705016378?ls=1&mt=8

Identifying Early Adopters for Emerging Digital Travel Services
https://www.cognizant.com/industries-resources/travel_and_hospitality/2016-Travel-Hospitality-Study.pdf

6 Ways to Build the Guest Experience Around Automation
https://www.hotelmanagement.net/operate/how-to-develop-a-better-guest-experience-around-automation